IEEE P1003.2a Draft 8 – December 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

5.35 vi — Screen-oriented (visual) display editor

This utility shall be provided on systems that both support the User Portability Utilities Option and define the {POSIX 2_CHAR _TERM } symbol. On other systems it is optional.

5.35.1 Synopsis

`vi [−rR][−c` *command* `][−t` *tagstring* `][−w` *size* `][`*file* `...]`

Obsolescent Version :

`vi [−rR][+`*command* `][−t` *tagstring* `][−w` *size* `][`*file* `...]`

5.35.2 Description

The `vi` (visual) utility is a screen-oriented text editor. The user can switch back and forth between `vi` and the line editor `ex` (see 5.10) and execute `ex` commands from within `vi`.

When using `vi`, the terminal screen acts as a window into the editing buffer. Changes made to the editing buffer shall be reflected in the screen display; the position of the cursor on the screen shall indicate the position within the editing buffer.

Certain block-mode terminals do not have all the capabilities necessary to support the complete `vi` definition. When these commands cannot be supported on such terminals, this condition shall not produce an error message such as ‘‘not an editor command’’ or report a syntax error. The implementation may either accept the commands and produce results on the screen that are the result of an unsuccessful attempt to meet the requirements of this standard or report an error describing the terminal-related deficiency.

5.35.3 Options

The `vi` utility shall conform to the utility argument syntax guidelines described in 2.10.2, except for the obsolescent +*command* ‘‘option.’’

The following options shall be supported by the implementation:

`−c` *command*
+*command* (Obsolescent.)
Begin editing by executing the specified `ex` command-mode command(s). As with normal `ex` command-line entries, the *command* option-argument can consist of multiple `ex` commands separated by vertical-line characters (|). The use of commands that enter input mode in this manner produces undefined results.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6258 `-r` Attempt to recover the named *file*s after an editor or system
6259 crash, after the editor has been terminated by a signal or after
6260 the use of a `pre` (`ex`) editor command.

6261 If no *file* operands are given, all other options, the **EXINIT** vari- 8
6262 able, and any `.exrc` files shall be ignored; a list of all recoverable 8
6263 files available to the invoking user shall be written; and `vi` shall 8
6264 exit without reading files or processing user commands. 8

6265 `-R` Set read-only mode, preventing accidental overwriting of the files. 8
6266 Any command that would write to a file shall require the `!` suffix 8
6267 (see, for example, 5.10.7.2.37) to be effective in this mode. 8

6268 `-t` *tagstring*
6269 Edit the file containing the specified *tagstring* and set the initial
6270 position within the edit buffer to the point of definition of the tag.
6271 (See `ctags` in 5.7.) The tags feature represented by `-t` *tagstring*
6272 and the `ta` command (see 5.10.7.2.32) is optional. It shall be pro- 8
6273 vided on any system that also provides a conforming implementa- 8
6274 tion of `ctags`; otherwise, the use of `-t` produces undefined 7
6275 results. 7

6276 `-w` *size* Set the value of the *window* editor option to *size*. See 5.10.7.5.28.

6277 If both the `-t` *tagstring* and `-c` *command* (or the obsolescent `+`*command*) options 8
6278 are given, the `-t` *tagstring* shall be processed first; i.e., the file containing the tag 8
6279 is selected by `-t` and then the command is executed. 8

6280 ### 5.35.4 Operands 8

6281 The following operand shall be supported by the implementation: 8

6282 *file* A pathname of a file to be edited. 8

6283 ### 5.35.5 External Influences 8

6284 #### 5.35.5.1 Standard Input 8

6285 The standard input consists of a series of commands and input text, as described 8
6286 in 5.35.7. 8

6287 #### 5.35.5.2 Input Files 8

6288 Input files shall be text files or files that would be text files except for an incom- 8
6289 plete last line that is not longer than {LINE_MAX} − 1 bytes in length and contains 8
6290 no NUL characters. The editing of other forms of files may optionally be allowed 8
6291 by `vi` implementations. 7

6292 The `.exrc` files (see 5.35.7) shall be text files consisting of `ex` commands. 8

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

By default, `vi` shall read lines from the files to be edited without interpreting any of those lines as any form of editor command.

5.35.5.3 Environment Variables

The following environment variables shall affect the execution of `vi`:

COLUMNS This variable shall override the system-selected horizontal screen size. See 2.6 for valid values and results when it is unset or null.

EXINIT This variable shall be interpreted to contain a list of `ex` commands that shall be executed on editor startup, before reading the first file. The list can contain multiple commands by separating them using a vertical-line (`|`) character. See also 5.35.7 for more details of the initialization phase.

HOME This variable shall be interpreted as a pathname of a directory that shall be searched for an editor startup file named `.exrc`; see 5.35.7 for details.

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_COLLATE This variable shall determine the locale for character collation information in regular expressions.

LC_CTYPE This variable shall determine the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files), the behavior of character classes within regular expressions, the classification of characters as upper- or lowercase letters, the case conversion of letters, and the detection of word boundaries.

LC_MESSAGES This variable shall determine the language in which messages should be displayed.

LINES This variable shall override the system-selected vertical screen size, used as the number of lines in a screenful and the vertical screen size in visual mode. See 2.6 for valid values and results when it is unset or null.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

PATH This variable shall determine the search path for the shell command specified in the editor commands `shell`, `read`, and `write` and the visual-mode command `!`; see the description of command search and execution in 3.9.1.1.

SHELL This variable shall be interpreted as the preferred command-line interpreter for use in `!`, `shell`, `read`, and other commands with an operand of the form `!`*string* . For the `shell` command the program shall be invoked with the single argument `-i`, for all others it shall be invoked with the two arguments `-c` and *string* . If this variable is null or not set, the `sh` (see 4.56) utility shall be used.

TERM This variable shall be interpreted as the name of the terminal type. If this variable is unset or null, an unspecified default terminal type shall be used.

5.35.5.4 Asynchronous Events

The following actions shall be taken upon receipt of signals:

SIGINT The current editor command shall be aborted. The editor shall prompt for another command.

SIGCONT The screen shall be refreshed (if in *visual* mode).

SIGHUP If the current buffer has changed since the last `e` or `w` command, `vi` shall attempt to save the current file in a state such that it can be recovered later by an `ex` or `vi -r` command.

The action taken for all other signals is unspecified.

5.35.6 External Effects

5.35.6.1 Standard Output

If standard output is a terminal device, it may be used for writing prompts to the user, for informational messages, and for writing lines from the file. If standard output is not a terminal device, undefined results occur.

5.35.6.2 Standard Error

Used only for diagnostic messages.

5.35.6.3 Output Files

The output from `vi` shall be text files that shall be identical to the input files if no changes have been made to those files by commands, with the exception that in all cases where a forced session termination (the `ex` command `q!`; see 5.10.7.2.23) has not been issued prior to any write of the file, a trailing `<newline>` shall be

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

added to the last line of the file if one was not present in the input.

5.35.7 Extended Description

Only the visual mode of the editor is described in this subclause. See 5.10 for additional editing capabilities used in `vi`.

The pathname of the file being edited by `vi` is the *current* file. The text of the file shall be read into a *buffer*, and all editing changes shall be performed in this buffer; changes shall have no effect on the file until the buffer is written out explicitly. (The use of the term *buffer* does not attempt to imply a specific implementation.) Lines in the buffer may each be limited to {LINE_MAX} bytes and an error message may be displayed if the limit is exceeded during editing.

The *alternate* pathname is the name of the last file mentioned in an editor command, or the previous current pathname if the last file mentioned became the current file. When the character `%` appears in a pathname entered as part of a command argument, the character shall be replaced by the current pathname; the character `#` shall be replaced by the alternate pathname. The characters `%` and `#` can be escaped by preceding them with a backslash.

During initialization, before the first file is read or any user commands from the terminal are processed, if the environment variable **EXINIT** is set, the editor shall execute the `ex` commands contained in that variable. If the variable is not set, `vi` shall attempt to read `ex` commands from the file `$HOME/.exrc` (the file `.exrc` in the directory referred to by the **HOME** environment variable). If and only if **EXINIT** or `$HOME/.exrc` sets the editor option `exrc`, `vi` finally shall attempt to read `ex` commands from a file `.exrc` in the current directory. In the event that **EXINIT** is not set and the current directory is the user's home directory, any `.exrc` file shall only be processed once. No `.exrc` file shall be read unless it is owned by the same user ID as the effective user ID of the process. After any `.exrc` files are processed, any commands specified by the `−c` option shall be processed.

After initialization, `vi` shall be in *command* mode; *input* mode can be entered by several commands used to insert or change text. In input mode, `<ESC>` can be used to return to command mode; other uses of `<ESC>` are described in 5.35.7.1.17. The results of entering `<newline>`s in input mode are affected by the setting of the `autoindent` editor variable; see 5.10.7.5.1.

The last (bottom) line of the screen shall be used to display the input for search commands (`/` and `?`), for `ex` commands (`:`), and system commands (`!`). It shall be also used to report errors or display informational messages. The editor shall prompt for input for commands by displaying the command (`/`, `?`, `:`, or `!`) at the beginning of the last line of the screen. All subsequent characters shall be then taken as input until a line terminator is entered.

An interrupt typed during text input, or during the input of a command on the bottom line, shall terminate the input (or cancel the command) and shall return the editor to command mode. During command mode, an interrupt shall alert the terminal. In general, an alert indicates an error (such as unrecognized key).

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Lines displayed on the screen containing only a tilde (~) shall indicate that the last line above them is the last line in the editing buffer (the ~ lines are past the end of the editing buffer).

There may be lines on the screen marked with an @. These indicate space on the screen not corresponding to lines in the file. This shall happen if there is space at the bottom of the screen for some text, but the next line of the buffer will not fit. (It may also happen on a terminal with limited local intelligence. In this case, these lines can be removed by entering a `control-R`, forcing the editor to refresh the screen.)

5.35.7.1 `vi` Command Descriptions

The cursor, in general, shall be placed on the current line and in the current column position as noted for each command described below. If the current line is not in the display window, then the display window shall be either scrolled or refreshed to cause the current line to be in the display window. If the screen is refreshed, the current line shall be positioned as close to the center of the display window as possible. If the current line is less than one-half window lines from the beginning of the editing buffer, the first line of the buffer shall be displayed on the first line of the display window. If the current line is less than one-half window lines from the end of the editing buffer, the remaining lines of the display window after the last line of the file shall contain only a ~ character in column position 1. If the screen is scrolled rather than refreshed, the current line shall be placed at the top of the display window if the current line is before the first line displayed. If the current line is after the last line displayed, the display window shall be scrolled up and the current line shall be placed on the last line of the display window.

As stated previously, the cursor is generally placed on the current column position of the current line. The one exception to this is if the current column position is beyond the end of the current line. In this case, the cursor shall be placed on the last character of the current line. However, the current column position value shall not be altered by this. Thus, if the current line changes to a longer line, the cursor shall be moved back out to the current column position or to the end of the line if it is shorter than the current column.

Unless otherwise specified, the commands shall be interpreted in command mode and have no special effect in input mode.

Some of the following command descriptions (such as `a`, `A`, `c`, etc.) move the cursor and enter input mode. The indications of *Current line* and *Current column* are for the cursor movement up to the beginning of input mode. Depending on the text inserted, the cursor will generally wind up in a totally different location by the time command mode is reentered. The cursor movement that accompanies the transition from input mode to command mode is described in 5.35.7.1.17.

The following symbols are used in this section to represent arguments to commands, to describe commands, or to specify the new values of the current line and column indicators following execution of the command.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

bigword A maximal sequence of nonblank characters preceded and followed by <blank> characters or the beginning or end of a line or the file.

buffer One of a number of named areas for saving text. Commands that change or delete text can be preceded by a buffer specification argument *buffer*. It is specified in the form ''"*c*'', where *c* represents the name of a buffer, one of the lowercase letters of the POSIX Locale. Specifying *buffer* shall cause the area of text affected by the command to be stored into the buffer as it was before the command took effect. This argument is also used on the ''put'' commands (p and P) to specify the buffer that shall provide the text to insert. When a command synopsis shows both [*buffer*] and [*count*] preceding the command letter, either can precede the other.

If the buffer name is specified in uppercase, and the buffer is to be modified (as with a deletion or yanking command) the buffer shall be appended to rather than being overwritten. If the buffer is not to be modified (as in a visual mode ''put'' commands) the buffer name can be specified in lowercase or uppercase with the same results. There shall be also one unnamed buffer, which is the repository for all text deleted (with the delete or visual mode d command) or yanked (with the yank or visual mode y command) when no buffer is specified.

There are also numbered buffers, 1 through 9, which shall be accessible only from visual mode. These buffers are special in that, in visual mode, when deleted text is placed in the unnamed buffer, it also shall be placed in buffer 1, the previous contents of buffer 1 shall be placed in buffer 2, and so on. Any text in buffer 9 shall be lost. Text that is yanked (or otherwise copied) into the unnamed buffer shall not modify the numbered buffers. Text cannot be placed directly into the numbered buffers although it can be retrieved from them by using a visual mode ''put'' command with the buffer name given as a number. When the *buffer* modifier is not used in the commands below, the unnamed buffer shall be the default.

column The (previous) value of the current column indicator.

count A positive integer used as an optional argument to most commands, either to give a size or a position (for display or movement commands), or as a repeat count (for commands that change text). This argument is optional and shall default to 1 unless otherwise noted in the individual command description.

end-of-line A description of the current column indicator value meaning that the current column indicator shall be set always to indicate the last character of the current line. Until the current column indicator is changed, the cursor shall be always positioned to the last

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6498 character of whatever line is current.

6499 *line* The (previous) value of the current line indicator.

6500 *nonblank* A description of the current column indicator value meaning that
6501 the current column indicator shall be set to the position of the
6502 first nonblank character of the current line. If the current line
6503 has no nonblank characters, the indicator shall be set to the posi-
6504 tion of the last character of the line.

6505 *motion* A command used as an optional trailing argument to some com-
6506 mands, indicating the extent of text to be affected by the com-
6507 mand. The *motion* argument can be either the command charac-
6508 ter repeated or a cursor movement command. In the former case,
6509 exactly the current line shall be affected. In the latter case, the
6510 region specified shall be from the current cursor position to just
6511 before the cursor position indicated by the motion command. If
6512 the command operates on lines only, then all the lines that fall
6513 partly or wholly within this region shall be affected. Otherwise,
6514 the exact region as described above shall be affected. The follow-
6515 ing commands are considered to be cursor motion commands:

6516 | | | | |
|---|---|---|---|
6517 | `$` | `^` | `e` | `M` |
6518 | `%` | `''` | `control-F` | `control-N` |
6519 | `''` | `'`*letter* | `F` | `N` |
6520 | `'`*letter* | `{` | `f` | `n` |
6521 | `(` | `\|` | `G` | `control-P` |
6522 | `)` | `}` | `control-H` | `<space>` |
6523 | `,` | `0` | `H` | `control-T` |
6524 | `-` | `control-B` | `control-J` | `T` |
6525 | `/` | `B` | `j` | `t` |
6526 | `;` | `b` | `k` | `control-U` |
6527 | `?` | `control-D` | `L` | `W` |
6528 | `[[` | `control-E` | `l` | `w` |
6529 | `]]` | `E` | `control-M` | `control-Y` |

6530 The optional *count* prefix available for some of the motion com- 7
6531 mands can be included and is considered part of the *motion* argu- 7
6532 ment. For example, in `c2w`, the `2w` is the *motion* argument. 7

6533 *previous context*
6534 A state set whenever a nonrelative motion command is executed
6535 in `vi`.

6536 *paragraph* An area of text that begins with either the beginning of a file, an
6537 empty line, or section boundary and continues until either an
6538 empty line, section boundary, or the end of the editing buffer.
6539 Additional paragraph boundaries can be defined by the `para-`
`graph` option. (See 5.10.7.5.12.)

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

section An area of text that starts with a line whose first character is either a `<form-feed>` or an open brace (`{`) and continues until the next section or the end of the editing buffer. The beginning and end of the editing buffer are also considered section boundaries. Additional section boundaries can be defined by the `sections` option. (See 5.10.7.5.18.)

sentence An area of text that begins with either the beginning of the editing buffer or the first nonblank character following the previous sentence, paragraph, or section boundary and continues until the end of the editing buffer or a period, exclamation point, or question mark (`.`, `!`, `?`) character followed by either an end of line or two `<space>`s. Any number of closing parentheses, brackets, or double-quote (`)`, `]`, `"`) characters can appear between the period, exclamation point, or question mark and the `<space>`s or end of line.

window The current value of the editor option `window`.

word In the POSIX Locale, `vi` recognizes two kinds of words:

— A maximal sequence of letters, digits and underscores, delimited at both ends by: characters other than letters, digits, or underscores; the beginning or end of a line; or the end of the editing buffer.

— A maximal sequence of characters other than letters, digits, underscores, or white space, delimited at both ends by: a letter, digit, underscore, or white space; the beginning or end of a line; or the end of the editing buffer.

! A character that can be appended to the command to modify its operation as detailed in the individual command descriptions.

After processing a command, if the current line and column indicators have been changed to specify a character not currently on the screen, the editor shall scroll or page the text to bring the needed character onto the screen. If the new current line is within *window* lines of the previous current line, and the terminal is capable of scrolling in the indicated direction, the text shall be scrolled the minimum number of lines necessary to present the needed character. Otherwise, the screen shall be redisplayed with the new current line positioned in the middle of the screen.

The following rules specify how exceptions shall be handled. When a movement command would cause a window to display beyond the beginning of the editing buffer, the window displayed is the window beginning with the first line of the editing buffer. When a movement command would cause a window to be displayed beyond the end of the editing buffer, the terminal shall be alerted and the command aborted. When the current line is empty, the cursor shall be placed in the first position of the line on the screen. When the current column indicator value is beyond the end of the current line, the cursor shall be placed on the last character of the current line, but the current column indicator shall be

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

unchanged.

The following rules specify how position exceptions shall be handled. When the current column indicator is at the beginning (end) of a line and would be set to a position before the beginning (after the end) of the current line, the command shall not be executed, the terminal shall be alerted, and the current position shall remain unchanged. Otherwise, when the current column indicator would be set to a position before the beginning (after the end) of the current line, it shall be set instead to the position of the first (last) character of that line. When the current line indicator would be set to a position before the beginning (after the end) of the editing buffer, the command shall not be executed, the terminal shall be alerted, and the current position shall remain unchanged.

5.35.7.1.1 Page backwards

Synopsis : [*count*] `control-B`

Page backward in the file, allowing two lines of overlap, by displaying the window starting at line [*line* − (*count* ∗ (*window* −2))]. 7

Current line : The last line displayed.

Current column : Move to the first nonblank character of the current line or the first character if the line is a blank line.

5.35.7.1.2 Scroll forward

Synopsis : [*count*] `control-D`

Scroll forward *count* lines in the file. If *count* is not specified, the same number of lines shall be scrolled as by the previous `control-D` or `control-U` command. On the first `control-D` or `control-U` command, the amount scrolled shall be one-half the number of lines in a screenful. See 5.10.7.5.17.

Current line : (*line* + amount scrolled).

Current column : Move to the first nonblank character of the current line or the first character if the line is a blank line.

5.35.7.1.3 Scroll forward by line

Synopsis : [*count*] `control-E`

Scroll forward *count* lines, leaving the current line and column as is if possible.

Current line : Unchanged unless the current line scrolls off the screen; otherwise, move to the first line displayed.

Current column : Unchanged unless the current line scrolls off the screen; otherwise, move to the first character if it is a blank line or the last character of the line if the position of the current column is further into the line than the position of the last column of the line.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6620 #### 5.35.7.1.4 Page forward

6621 *Synopsis* : [*count*] `control-F`

6622 Page forward in the file, allowing two lines of overlap, by displaying the window
6623 starting at line [*line* + (*count* ∗ (*window* −2))]. 7

6624 *Current line* : The first line displayed.

6625 *Current column* : Move to the first nonblank character of the current line or the
6626 first character if the line is a blank line.

6627 #### 5.35.7.1.5 Display information

6628 *Synopsis* : `control-G`

6629 Display an informational message listing the current pathname, current line,
6630 number of lines, and other unspecified information, as does the `ex` command *file* .

6631 *Current line* : Unchanged.

6632 *Current column* : Unchanged.

6633 #### 5.35.7.1.6 Move cursor backwards

6634 *Synopsis* : [*count*] `control-H`
6635 *Synopsis* : [*count*] `h`

6636 Move the cursor back *count* characters on the current line. No characters shall be
6637 erased from the screen by this command. In input mode, `control-H` shall have
6638 the exact same function.

6639 *Current line* : Unchanged.

6640 *Current column* : Set to (*column* − the number of columns occupied by *count* char-
6641 acters ending with the previous current column) or the beginning of the line if
6642 *count* is greater than the number of characters preceding the current character in
6643 the current line.

6644 #### 5.35.7.1.7 Move down in column

6645 *Synopsis* : [*count*] `control-J`
6646 *Synopsis* : [*count*] `j`
6647 *Synopsis* : [*count*] `control-N`

6648 Move the cursor down *count* lines without changing the current column.

6649 *Current line* : *current line* + *count* . Unchanged if *count* is greater than the number
6650 of lines in the buffer following the current line.

6651 *Current column* : Unchanged.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6652 #### 5.35.7.1.8 Clear and redisplay

6653 *Synopsis* : `control-L`

6654 Clear and redisplay the screen.

6655 *Current line* : Unchanged.

6656 *Current column* : Unchanged.

6657 #### 5.35.7.1.9 Move cursor down to nonblank

6658 *Synopsis* : [*count*] `control-M`
6659 *Synopsis* : [*count*] `+`

6660 Move the cursor down *count* lines to the first nonblank character of that line.

6661 *Current line* : *line* + *count* .

6662 *Current column* : Move to the first nonblank character of the current line or the
6663 first character if the line is a blank line.

6664 #### 5.35.7.1.10 Move up in column

6665 *Synopsis* : [*count*] `control-P`
6666 *Synopsis* : [*count*] `k`

6667 Move the cursor up *count* lines without changing the current column.

6668 *Current line* : *line* − *count* .

6669 *Current column* : Unchanged.

6670 #### 5.35.7.1.11 Redraw screen

6671 *Synopsis* : `control-R`

6672 Redraw the current screen. If any lines have been deleted from the logical screen
6673 and flagged as deleted on the terminal using the ‘‘@’’convention (see 5.35.7), they
6674 shall be deleted, and the logical screen shall be redisplayed. Lines flagged with @
6675 because they do not fit on the terminal display shall not be affected.

6676 *Current line* : Unchanged.

6677 *Current column* : Unchanged.

6678 #### 5.35.7.1.12 Move forward with tabs

6679 *Synopsis* : `control-T`

6680 Move the cursor forward *shiftwidth* (see 5.10.7.5.20) positions in insert mode by
6681 inserting `<tab>` characters, followed by space characters, as necessary, if the
6682 current cursor position is at the beginning of a line or preceded only by `<blank>`s.

6683 *Current line* : Unchanged.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Current column : *column* + *shiftwidth* .

5.35.7.1.13 Scroll backward

Synopsis : [*count*] `control-U`

Scroll backward *count* lines in the file. If *count* is not specified, the same number of lines shall be scrolled as by the previous `control-D` or `control-U` command. On the first `control-D` or `control-U` command, the amount scrolled shall be the value of the `scroll` editor option; see 5.10.7.5.17.

Current line : (*line* − amount scrolled).

Current column : Move to the first nonblank character of the current line or the first character if the line is a blank line.

5.35.7.1.14 Escape next character

Synopsis : `control-V` *character*
Synopsis : `control-Q` *character* 8

Allow the entry of a subsequent *character* , removing any special meaning to the editor it has in input mode (e.g., <ESC>). The character ^ shall be displayed in the current cursor position until the subsequent character is typed, which then replaces the character ^.

Current line : Unchanged.

Current column : Unchanged.

5.35.7.1.15 Delete word

Synopsis : `control-W`

Delete the word preceding the cursor (including any <blank>s between the end of the word and the current cursor position) in input mode by moving the cursor back to the beginning of the word. No characters shall be erased from the screen by this command. Only text entered during the current input mode session and on the current line can be deleted with this command.

Current line : Unchanged.

Current column : Moved back as described above.

5.35.7.1.16 Scroll backward by line

Synopsis : [*count*] `control-Y`

Scroll backward *count* lines, leaving the current line and column as is, if possible.

Current line : Unchanged unless the current line scrolls off the screen; otherwise, the last line displayed.

Current column : Unchanged unless the current line scrolls off the screen; otherwise, move to the first character if it is a blank line or the last character of the line if the position of the current column is further into the line than the position

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

of the last column of the line.

5.35.7.1.17 Terminate command or input mode

Synopsis : <ESC>

The effects of <ESC> depend on the mode and the command being entered, as follows:

— In command mode:

- If a line-oriented command (/, ?, :, or !) is being entered, terminate the line-oriented command and execute it.
- If only part of a command has been entered, cancel the partial command. A command is not considered to be partially entered until at least one non-*count* character has been entered. For example, '33c<ESC>'' is partially entered and silently canceled, whereas '33<ESC>''alerts the terminal (see the next item).
- Otherwise, alert the terminal.

— In input mode: Terminate input mode and return to command mode. At this point any text that was deleted in input mode, but that has not yet been erased from the screen, shall be erased from the screen; the current line shall be redisplayed to match exactly the changes made in input mode.

Current line : When a line-oriented command is being terminated, the current line indicator shall be set as given for that command. Otherwise, unchanged.

Current column : When a line-oriented command is being terminated, the current column indicator shall be set as given for that command; When terminating input mode, if the cursor is not at the beginning of a line, the current column position shall be set to (*column* − the width of the last character inserted). Otherwise, unchanged.

5.35.7.1.18 Move cursor forward

Synopsis : [*count*] <space>
Synopsis : [*count*] l (ell)

Move the cursor forward *count* characters without changing the current line.

Current line : Unchanged.

Current column : (*current column* + the width of the next *count* characters) or the end of the line if *count* is greater than the number of characters in the line following the current column.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.35.7.1.19 Replace text with results from shell command

Synopsis : [*count*] ! *motion shell-command(s)* <newline>

Pass the lines specified by *count* and *motion* as standard input to a shell command, for the program named in the **SHELL** environment variable, and replace those lines with the standard output of the shell command. An implementation may consider a *motion* command that does not result in an integral number of lines to be an error. After the *motion* is entered, the text of the command shall be prompted for on the last line of the display as described in 5.35.7. A warning shall be issued if the buffer has been changed since the last write.

Within the text of *command* , % and # shall be expanded as pathnames (the current and alternate pathnames, respectively), and ! shall be replaced with the text of the previous ! or :! command. (Thus, !! shall repeat the previous ! command.) If no ! or :! command has yet been executed, an informational message shall be displayed stating the problem.

The special meanings of %, #, and ! can be overridden by escaping them with a backslash character. This command shall be affected by the editor options autowrite and writeany; see 5.10.7.5 for details.

Current line : The first line in *range* .

Current column : The first column of the replaced text.

5.35.7.1.20 Move cursor to end-of-line

Synopsis : [*count*] $

Move the cursor forward to the end of a line. The *count* argument specifies the number of lines, including the current line, to move forward.

Current line : *line* + *count* − 1.

Current column : *end-of-line* .

5.35.7.1.21 Move to matching character

Synopsis : %

Move the cursor to the parenthesis or curly brace matching the parenthesis or curly brace at the current position or on the current line forward from the current position. Matching shall be determined as follows: for a left parenthesis (respectively curly brace) the editing buffer shall be searched forward until either a left or right parenthesis is found. If a left parenthesis is found, a counter is incremented by one, and if a right parenthesis is found, a counter is decremented by one, and the search continues (until the end of the editing buffer is found). If a right parenthesis is found when the count is less than or equal to zero, it is the matching parenthesis. For a right parenthesis (respectively curly brace) the editing buffer shall be searched backward (to the beginning of the editing buffer), and each right parenthesis increments the counter, and each left parenthesis decrements it. When searching for parentheses, curly braces shall not be counted, and vice versa. If there is no matching character, the terminal shall be alerted and

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

the current position shall remain unchanged.

Current line : Move to the line containing the matching character, as described above.

Current column : Move to the column containing the matching character, as described above.

5.35.7.1.22 Repeat substitution

Synopsis : &

Repeat the previous substitution command (equivalent to the `ex` & command; see 5.10.7.2.44) using the current line as the target. Flags specified on the previous substitution command shall be ignored by this command.

Current line : Unchanged.

Current column : Unchanged if the previous current column indicator value was *end-of-line* ; otherwise, move to the first nonblank character of the current line or the first character if the line is a blank line.

5.35.7.1.23 Return to previous context (at beginning of line)

Synopsis : ´´
Synopsis : ´ *letter*

Return to the previous context when followed by a ´ character, if the context still exists, placing the cursor at the first nonblank character of the line or the last character if the line is a blank line. If the previous context no longer exists, the ´ command shall have no effect. When followed by a lowercase letter from the POSIX Locale, return to the line marked with that letter (see the `m` command), at the first nonblank character in the line.

When used with an operator such as `d` to specify an extent of text, the operation takes place over complete lines. (See also 5.35.7.1.24.)

Current line : Move to the line from the previous context.

Current column : Move to the first nonblank character of the current line or the last character if the line is a blank line.

5.35.7.1.24 Return to previous context

Synopsis : ``
Synopsis : ` *letter*

When followed by a ` character, return to the previous context, placing the cursor at the character position marked. (The previous context shall be set whenever a nonrelative move is made.) When followed by a lowercase letter from the POSIX Locale, return to the line marked with that letter (see the `m` command), at the character position marked.

When used with an operator such as `d` to specify an extent of text, the operation takes place from the exact marked place to the current position within the line.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6832 (See also 5.35.7.1.23.)

6833 *Current line* : Move to the line from the previous context.

6834 *Current column* : Set to the position of the character marked from the previous
6835 context.

6836 **5.35.7.1.25 Return to previous section**

6837 *Synopsis* : `[[`

6838 Back up to the previous section boundary. This command shall be affected by the
6839 `sections` option; see 5.10.7.5.18. Note that the brackets in this command are
6840 part of the command syntax itself and are not metacharacters.

6841 *Current line* : Move to the previous line that is a section boundary or to the first
6842 line of the editing buffer if no more section boundaries exist preceding the current
6843 line.

6844 *Current column* : Move to the first nonblank character of the current line or the
6845 last character if the line is a blank line.

6846 **5.35.7.1.26 Move to next section**

6847 *Synopsis* : `]]`

6848 Move forward to a section boundary. This command shall be affected by the `sec-`
6849 `tions` option; see 5.10.7.5.18. Note that the brackets in this command are part of
6850 the command syntax itself and are not metacharacters.

6851 *Current line* : Move to the next line that is a section boundary or to the last line of
6852 the editing buffer if no more section boundaries exist following the current line.

6853 *Current column* : Move to the first nonblank character of the current line or the
6854 last character if the line is a blank line.

6855 **5.35.7.1.27 Move to first nonblank position on current line**

6856 *Synopsis* : `^`

6857 Move to the first nonblank position on the current line.

6858 *Current line* : Unchanged.

6859 *Current column* : Move to the first nonblank character of the current line or the
6860 last character if the line is a blank line.

6861 **5.35.7.1.28 Move back to beginning of sentence**

6862 *Synopsis* : `[`*count*`] (`

6863 Move backward to the beginning of a sentence. A *count* shall move back that
6864 many sentences.

6865 *Current line* : Move to the line containing the beginning of the sentence.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Current column : Move to the first nonblank character of the sentence.

5.35.7.1.29 Move forward to beginning of sentence

Synopsis : [*count*])

Move forward to the beginning of a sentence. A *count* shall move forward that many sentences.

Current line : Move to the line containing the beginning of the sentence.

Current column : Move to the first nonblank character of the sentence.

5.35.7.1.30 Move back to preceding paragraph

Synopsis : [*count*] {

Move back to the beginning of the preceding paragraph. A *count* specifies the number of paragraphs to move backward. This command shall be affected by the `paragraph` option; see 5.10.7.5.12.

Current line : Move to the line containing the beginning of the previous paragraph.

Current column : Move to the first nonblank character of the current line.

5.35.7.1.31 Move forward to next paragraph

Synopsis : [*count*] }

Move forward to the beginning of the next paragraph. A *count* specifies the number of paragraphs to move forward. This command shall be affected by the `paragraph` option; see 5.10.7.5.12.

Current line : Move to the line containing the beginning of the next paragraph.

Current column : Move to the first nonblank character of the current line.

5.35.7.1.32 Move to specific column position

Synopsis : [*count*] |

Place the cursor in the column position identified by *count* (if that position exists on the line). If *count* is omitted, it shall default to 1. If the length of the current line is less than *count* , the cursor shall be placed at the end of the current line. If the column position is spanned by a `<tab>` or a wide character, the cursor shall be placed on the character following the *count* -th column position.

Current line : Unchanged.

Current column : Move to the column position represented by *count* .

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6897 **5.35.7.1.33 Reverse find character**

6898 *Synopsis* : [*count*] `,`

6899 Reverse the last `f`, `F`, `t`, or `T` command, looking the other way in the current line.
6900 A *count* shall be equivalent to repeating the search that many times.

6901 *Current line* : Unchanged.

6902 *Current column* : Move to the first column position of the next character found.

6903 **5.35.7.1.34 Move to first nonblank of previous line**

6904 *Synopsis* : [*count*] –

6905 Move to the first nonblank character in the previous line. A *count* specifies how
6906 many lines to move back.

6907 *Current line* : *line* − *count*

6908 *Current column* : Move to the first nonblank character of the current line or the
6909 last character if the line is a blank line.

6910 **5.35.7.1.35 Repeat**

6911 *Synopsis* : [*count*] `.`

6912 Repeat the last command that changed the buffer. A *count* shall be passed on to
6913 the command being repeated.

6914 *Current line* : Dependent on the results of the execution of the last command that
6915 changed the buffer.

6916 *Current column* : Dependent on the results of the execution of the last command
6917 that changed the buffer.

6918 **5.35.7.1.36 Find regular expression**

6919 *Synopsis* : `/`

6920 Prompt the user to enter a string on the last line on the screen, interpret it as a
6921 regular expression (see 5.10.7.3), and scan forward for the next occurrence of a
6922 matching string. The search shall begin when a `<newline>` or an `<ESC>` is
6923 entered to terminate the pattern; it can be prematurely terminated with an inter-
6924 rupt. 7

6925 When used with an operator to specify an extent of text, the defined region is from
6926 the current cursor position to the beginning of the matched string. Whole lines
6927 can be specified by giving an offset from the matched line (using a closing `/` fol-
6928 lowed by a +*n* or −*n*).

6929 Regular expressions are described in `ex`; see 5.10.7.3.

6930 *Current line* : Move to the line in which the first (or next) regular expression
6931 match occurred.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Current column : Move to the column of the first character of the matched string.

5.35.7.1.37 Move to first character in line

Synopsis : `0` (zero)

Move to the first character on the current line. (The zero shall not be interpreted as a command when it is preceded by a nonzero digit.)

Current line : Unchanged.

Current column : Column position 1.

5.35.7.1.38 Execute an `ex` command

Synopsis : `:`

Begin an `ex` command. The `:` character, as well as the entered command, shall be displayed on the bottom line. The command shall be executed when the input is terminated by entering a `<newline>` or an `<ESC>`. Typing a `control-V` (or `control-Q`) character shall escape the following character, allowing its literal value to be entered.

NOTE: This special property of `control-V` and `control-Q` is only effective in visual or open modes and in `ex` commands called from visual/open mode with the `:` command.

If the `ex` command causes the screen to be scrolled or causes an escape to the shell, `vi` shall display a message indicating that it is waiting for a `<newline>` character. When a `<newline>` or an `<ESC>` character is entered, the screen shall be refreshed.

Current line : Dependent on the results of the execution of the `ex` command.

Current column : Dependent on the results of the execution of the `ex` command.

5.35.7.1.39 Repeat find

Synopsis : **[***count* **]** `;`

Repeat the last character find using `f`, `F`, `t`, or `T`. A *count* shall move the cursor to (or next to) the *count* -th occurrence of the character.

Current line : Unchanged.

Current column : Move to the column containing the character being searched for. If no matching character is found, it shall be unchanged.

5.35.7.1.40 Shift left

Synopsis : **[***count* **]** `<` **[***motion* **]**

Shift lines left one *shiftwidth* (see 5.10.7.5.20). The command can be followed by a motion command to specify lines. A *count* shall be passed through to the motion command.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

When *motion* is **<**, it shall shift the current line (or *count* lines starting at the current one).

Current line : Unchanged.

Current column : Move to the first nonblank character of the line or the last character if the line is a blank line.

5.35.7.1.41 Shift right

Synopsis : **[** *count* **]** **>** **[** *motion* **]**

Shift lines right one *shiftwidth* (see 5.10.7.5.20). The command can be followed by a motion command to specify lines. A *count* shall be passed through to the motion command.

When *motion* is **>**, it shall shift the current line (or *count* lines starting at the current one).

Empty lines shall not be changed.

Current line : Unchanged.

Current column : Move to the first nonblank character of the line or the last character if the line is a blank line.

5.35.7.1.42 Scan backwards for regular expression

Synopsis : **?**

Scan backwards, the reverse of **/**. See 5.35.7.1.36.

Current line : Move to the line in which the next succeeding regular expression match occurred.

Current column : Move to the column of the first character of the matched string.

5.35.7.1.43 Execute

Synopsis : *@buffer*

Execute each line of the named buffer as one or more **vi** commands (including **ex** commands, as described in 5.35.7.1.38). If the buffer name is **@**, then the last buffer executed shall be used; if there is no last buffer, an error shall occur. The text of a macro may contain an **@** character calling another macro; however, recursively calling the same macro produces undefined behavior. A **vi** error shall terminate all currently executing macros. All changes made during a macro call shall be treated as a unit and can be undone with a single **u** command.

Current line : Dependent on the results of the execution of the **vi** command(s).

Current column : Dependent on the results of the execution of the **vi** command(s).

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6999 #### 5.35.7.1.44 Reverse case

7000 *Synopsis* : [*count*] ~

7001 Reverse the case of the *count* character(s) beginning at the current cursor posi-
7002 tion. Lowercase alphabetic characters shall be changed to uppercase and upper-
7003 case characters changed to lowercase. This command shall have no effect on
7004 nonalphabetic characters.

7005 *Current line* : Unchanged.

7006 *Current column* : Move to the next column unless it is on the last character of the
7007 line, in which case it shall remain unchanged.

7008 #### 5.35.7.1.45 Append

7009 *Synopsis* : [*count*] a

7010 Enter input mode, appending the entered text after the current cursor position. A
7011 *count* shall cause the inserted text up to the first <newline> to be replicated that 8
7012 many times; if a <newline> appears in the inserted text, only one occurrence of it 8
7013 and any characters following it shall be inserted. For example, if the current line 8
7014 is abc, the command 8

7015 ```
03afoo<newline>bar
```
8

7016 changes that line to 8

7017 ```
afoofoofoo
```
8
7018 ```
barbc
```
8

7019 *Current line* : Unchanged. 7

7020 *Current column* : *column* + 1. 7

7021 #### 5.35.7.1.46 Append at end of line

7022 *Synopsis* : [ *count* ] A

7023 Append *count* copies of the input text at the end of the current line, equivalent to
7024 $[ *count* ] a. See the a command in 5.35.7.1.45.

7025 *Current line* : Unchanged. 8

7026 *Current column* : See 5.35.7.1.45.

7027 #### 5.35.7.1.47 Move backward to preceding word

7028 *Synopsis* : [ *count* ] b

7029 Move the cursor backward to the beginning of a word by repeating the following
7030 algorithm *count* times: If the current position is at the beginning of a word, the 7
7031 current position shall move to the first character of the preceding word. Other-
7032 wise, the current position shall move to the first character of the word at the
7033 current position. If no preceding word exists on the current line, the current posi-
7034 tion shall move to the first character of the last word on the first preceding line

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.
```

that contains a word. For this command, an empty or blank line shall be considered to contain exactly one word.

Current line : Set to the line containing the word selected.

Current column : Set to the first character of the word selected.

5.35.7.1.48 Move backward to preceding bigword

Synopsis : [*count*] B

Move the cursor backward to the beginning of a bigword by repeating the following algorithm *count* times: If the current position is at the beginning of a bigword or the character at the current position cannot be part of a bigword, the current position shall move to the first character of the preceding bigword. Otherwise, the current position shall move to the first character of the bigword at the current position. If no preceding bigword exists on the current line, the current position shall move to the first character of the last bigword on the first preceding line that contains a bigword. For this command, an empty or blank line is considered to contain exactly one bigword.

Current line : Set to the line containing the bigword selected.

Current column : Set to the first character of the bigword selected.

5.35.7.1.49 Change

Synopsis : [*buffer*] [*count*] c *motion*

Delete the specified region of text and enter input mode to replace it with the entered text. If more than part of a single line is affected, the deleted text shall be saved in the numeric buffers. If only part of the current line is affected, then the last character to be deleted shall be marked with a $. A *count* shall be passed through to the motion command.

Current line : The current line and column position are dependent on the *motion* command following the c. For example, cw changes a word, c[[changes from the beginning of the current section, etc.

Current column : See *Current line* , above

5.35.7.1.50 Change to end-of-line

Synopsis : [*buffer*] [*count*] C

Change text from the current position through the end of line (*line* + *count* − 1); equivalent to [*buffer*] [*count*] c$.

Current line : See the c command in 5.35.7.1.49.

Current column : See 5.35.7.1.49.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.35.7.1.51 Delete

Synopsis : [*buffer*] [*count*] d *motion*

Delete the specified region of text. If more than part of a line is affected, the text shall be saved in the numeric buffers as well as any named [*buffer*]. A *count* shall be passed through to the motion command.

Current line : The current line and column position are dependent on the *motion* command following the d. For example, dw deletes a word, d$ deletes to the end of the line, etc.

Current column : See *Current line* , above.

5.35.7.1.52 Delete to end-of-line

Synopsis : [*buffer*] [*count*] D

Delete the text from the current position to the end of the current line; equivalent to [*buffer*] [*count*] d$.

Current line : Unchanged.

Current column : Set to maximum of *column* − 1 or 1.

5.35.7.1.53 Move to end of word

Synopsis : [*count*] e

Move the cursor forward to the end of a word, by repeating the following algorithm *count* times: If the current position is the end of a word, the current position shall move to the last character of the following word. Otherwise, the current position shall move to the last character of the word at the current position. If no succeeding word exists on the current line, the current position shall move to the last character of the first word on the next following line that contains a word. For this command, an empty or blank line is considered to contain exactly one word.

Current line : Set to the line containing the word selected.

Current column : Set to the last character of the word selected.

5.35.7.1.54 Move to end of bigword

Synopsis : [*count*] E

Move the cursor forward to the end of a bigword, by repeating the following algorithm *count* times: If the current position is the end of a bigword or the character at that position cannot be part of a bigword, the current position shall move to the last character of the following bigword. Otherwise, the current position shall move to the last character of the bigword at the current position. If no succeeding bigword exists on the current line, the current position shall move to the last character of the first bigword on the next following line that contains a bigword.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

For this command, an empty or blank line is considered to contain exactly one bigword. 7

Current line : Set to the line containing the bigword selected. 7

Current column : Set to the last character of the bigword selected. 7

5.35.7.1.55 Find character in current line (forward)

Synopsis : [*count*] `f` *character*

Scan the rest of the current line for the single character *character* and move the cursor to it if it is found. A *count* shall repeat the find that many times.

Current line : Unchanged.

Current column : Set to the column position containing the character that was scanned for. Unchanged if insufficient characters were found to satisfy the *count* .

5.35.7.1.56 Find character in current line (reverse)

Synopsis : [*count*] `F` *character*

Scan backwards in the current line for the single character *character* , moving the cursor to it if it is found. A *count* shall be equivalent to repeating the search that many times. If the search is unsuccessful, the terminal shall be alerted and the current column position shall remain unchanged.

Current line : Unchanged.

Current column : Set to the last character found moving backwards.

5.35.7.1.57 Move to line

Synopsis : [*count*] `G`

Move the cursor to line *count* , or to the last line of the editing buffer if *count* is not specified.

Current line : *count* , if specified; otherwise, the last line.

Current column : Move to the first nonblank character of the current line or the last character if the line is a blank line.

5.35.7.1.58 Move to top of screen

Synopsis : [*count*] `H`

Move the cursor to the line (*count* − 1) lines from the top of the current window (i.e., the *count* -th line, one-based, currently displayed).

Current line : *count* on the screen, as described above.

Current column : Move to the first nonblank character of the current line or the last character if the line is a blank line.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.35.7.1.59 Insert before cursor

Synopsis : [*count*] i

Enter input mode, inserting the entered text before the cursor. (See also the a command in 5.35.7.1.45.) A *count* shall cause the inserted text up to the first <newline> to be replicated that many times; if a <newline> appears in the inserted text, only one occurrence of it and any characters following it shall be inserted.

Current line : Unchanged.

Current column : Unchanged.

5.35.7.1.60 Insert at beginning of line

Synopsis : [*count*] I

Enter input mode at the beginning of a line. This command shall behave identically to the ^[*count*]i command. See 5.35.7.1.59.

Current line : Unchanged.

Current column : Move to the first nonblank character of the current line or the last character if the line is a blank line.

5.35.7.1.61 Join

Synopsis : [*count*] J

Join the current line with the next one, supplying appropriate spacing (in the POSIX Locale): one <space> between words, two <space>s after a period, and no <space>s at all when the last character of the first line is a <blank> or when the first character of the next line is). In the case of lines ending with <blank>s, the <blanks> shall be retained, no spaces shall be added, and the lines shall be joined together. A *count* shall cause that many lines (minimum 2) to be joined, rather than 2. A count of 1 shall be treated as 2.

Current line : Unchanged.

Current column : Move to the character after the last character of the original line.

5.35.7.1.62 Move to bottom of screen

Synopsis : [*count*] L

Move the cursor to the first nonblank character of the last line on the screen. A *count* shall move to that line counting from the bottom. When used with an operator, whole lines shall be affected.

Current line : *last line* − *count*

Current column : Move to the first nonblank character of the current line or the last character if the line is a blank line.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.35.7.1.63 Mark position

Synopsis : `m` *letter*

Mark the current position of the cursor with a single POSIX Locale lowercase letter *letter*. The exact position is referred to by `` `letter ``; the line is referred to by `'letter`.

Current line : Unchanged.

Current column : Unchanged.

5.35.7.1.64 Move to middle of screen

Synopsis : `M`

Move the cursor to the middle line on the screen, at the first nonblank position on the line.

Current line : Move to a line approximately in the middle of the display window.

Current column : Move to the first nonblank character of the current line or the last character if the line is a blank line.

5.35.7.1.65 Repeat regular-expression find (forward)

Synopsis : `n`

Repeat the last `/` or `?` scanning command.

Current line : Set to the line containing the character string found by the scan. Unchanged if no match was found.

Current column : Move to the first character of the character string found by the scan. Unchanged if no match was found.

5.35.7.1.66 Repeat regular-expression find (reverse)

Synopsis : `N`

Scan for the next match of the last pattern given to `/` or `?`, but in the reverse direction; this is the reverse of `n`.

Current line : Move to the matched line if a match is found or unchanged if no match is found.

Current column : Move to the first column position of the matched string or unchanged if no match is found.

5.35.7.1.67 Insert empty line below

Synopsis : `o`

Open a line below the current line and enter input mode.

Current line : The newly-created line, (*line* +1).

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Current column : If the editor option `autoindent` is not set, move to the first column; otherwise, as in the description of `autoindent` in `ex` (see 5.10.7.5.1).

5.35.7.1.68 Insert empty line above

Synopsis : `O`

Open a new line above the current line and enter input mode.

Current line : Unchanged.

Current column : If the editor option `autoindent` is not set, move to the first column; otherwise, as described in the description of `autoindent` in `ex`.

5.35.7.1.69 Put from buffer following

Synopsis : [*buffer*] `p` 7

Insert text from buffer *buffer* following the current column or current line, as described for the `P` command (see 5.35.7.1.70). If *buffer* is omitted, the unnamed buffer shall be used. 7

Current line : (*Current line* +1) if the buffer contains whole lines; otherwise, unchanged.

Current column : First nonblank character of the inserted text if the buffer contained whole lines; otherwise, the last character of the inserted text.

5.35.7.1.70 Put from buffer before

Synopsis : [*buffer*] `P`

Put the last deleted text back before/above the cursor. The text shall go back as whole lines above the cursor if it was deleted as whole lines. Otherwise, the text shall be inserted just before the cursor.

The command can be preceded by a named buffer specification ("*x*), to retrieve the contents of the buffer.

Current line : Unchanged.

Current column : Move to the last column position of the inserted characters. If the buffer contains whole lines, the current column shall be the first nonblank character of the inserted characters.

5.35.7.1.71 Enter `ex` mode

Synopsis : `Q`

Quit from `vi` and enter `ex` command mode.

Current line : Unchanged.

Current column : Unchanged.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

7240 **5.35.7.1.72 Replace character**

7241 *Synopsis* : [*count*] r *character*

7242 Replace a character on the screen with the character entered. If *count* is
7243 specified, the single character entered shall replace the current character and
7244 each of the next *count* − 1 characters in the line. (For example, 7rx changes the
7245 next seven characters to be xxxxxxx). Unlike the R command (see 5.35.7.1.73),
7246 entering a <newline> shall replace the character with a <newline>; however,
7247 when used with *count* , the next *count* characters in the line shall be replaced by a
7248 single <newline>.

7249 *Current line* : Unchanged unless the replacement character is a <newline>, in
7250 which case the current line shall be incremented by 1.

7251 *Current column* : Set to the last column position occupied by the replacement
7252 character, unless the replacement character is a <newline>, in which case the
7253 current column position shall be 1.

7254 **5.35.7.1.73 Replace characters**

7255 *Synopsis* : R

7256 Replace characters on the screen with characters entered. If the end of the exist-
7257 ing line is encountered, it shall be as if insert mode was entered at that point. If a
7258 <newline> is entered before the end of the existing line, it shall be entered into
7259 the editing buffer as if it was inserted, and replacement shall continue on the
7260 newly created line. No other line but the one in which the R command was given
7261 shall be affected.

7262 *Current line* : Set to the line at the end of the replaced text.

7263 *Current column* : Set to the end of the replaced text, if text was replaced. If no 7
7264 text was replaced, set to the column position of the character after which text 7
7265 would have been replaced if that character is on the current line; otherwise, 1. 7

7266 **5.35.7.1.74 Substitute character**

7267 *Synopsis* : [*buffer*] [*count*] s 8

7268 Substitute *count* characters in the current line starting with the current column
7269 position; equivalent to [*buffer*] [*count*] c<space>. 8

7270 *Current line* : Unchanged.

7271 *Current column* : Unchanged.

7272 **5.35.7.1.75 Substitute lines**

7273 *Synopsis* : [*buffer*] [*count*] S 8

7274 Change whole lines (same as cc). A *count* shall change that many lines; *count*
7275 lines shall be deleted and vi shall enter input mode.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Current line : Unchanged.

Current column : Set to column position 1 or to the position indicated by the previous line if `autoindent` is set.

5.35.7.1.76 Move cursor to before character (forward)

Synopsis : [*count*] `t` *character*

Move the cursor forward within the current line to the character position just before a subsequent occurrence of *character* . A *count* shall place the cursor just before the *count* -th occurrence of the character when searching the line forwards. If the search is unsuccessful, the terminal shall be alerted and the cursor position shall remain unchanged.

Current line : Unchanged.

Current column : Move to the position before an occurrence of *character* following *column* (noninclusive).

5.35.7.1.77 Move cursor to after character (reverse)

Synopsis : [*count*] `T` *character*

Scan backwards in the current line for the single character *character* and if found, place the cursor just after that character. A *count* shall place the cursor just after the *count* -th occurrence of the character when searching the line backwards. If *count* -th occurrences of the character do not exist, the terminal shall be alerted and the current column position shall remain unchanged.

Current line : Unchanged.

Current column : Set to the column after the scanned character. Unchanged if the character was not found.

5.35.7.1.78 Undo

Synopsis : `u`

Reverse the last change made to the current buffer. If repeated, the command shall alternate between these two states, thus is its own inverse. When used after an insert that inserted text on more than one line, the lines shall be saved in the numeric named buffers.

Current line : Move to the position of the first line changed, if the reversal affects only one line or represents an addition or change; otherwise, move to the line preceding the deleted text.

Current column : Move to the first nonblank character on the updated current line.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

7310 #### 5.35.7.1.79 Undo current line

7311 *Synopsis* : `U`

7312 Restore the current line to its state before the cursor was last moved to it.

7313 *Current line* : Unchanged.

7314 *Current column* : Set to column position 1 or to the position indicated by the pre-
7315 vious line if `autoindent` is set.

7316 #### 5.35.7.1.80 Move to beginning of word

7317 *Synopsis* : [*count*] `w`

7318 Move the cursor forward to the beginning of a word by repeating the following
7319 algorithm *count* times: If the current position is at the beginning of a word, the 7
7320 current position shall move to the first character of the next word. If no subse-
7321 quent word exists on the current line, the current position shall move to the first
7322 character of the first word on the first following line that contains a word.

7323 For this command, an empty or blank line is considered to contain exactly one 7
7324 word. 7

7325 *Current line* : Set to the line containing the word selected. 7

7326 *Current column* : Set to the first character of the word selected. 7

7327 #### 5.35.7.1.81 Move to beginning of bigword

7328 *Synopsis* : [*count*] `W`

7329 Move the cursor forward to the beginning of a bigword by repeating the following
7330 algorithm *count* times: If the current position is within a bigword or the character
7331 at that position cannot be part of a bigword, the current position shall move to the
7332 first character of the next bigword. If no subsequent bigword exists on the
7333 current line, the current position shall move to the first character of the first big-
7334 word on the first following line that contains a bigword. For this command, an 7
7335 empty or blank line is considered to contain exactly one bigword. 7

7336 *Current line* : Set to the line containing the bigword selected.

7337 *Current column* : Set to the first character of the bigword selected.

7338 #### 5.35.7.1.82 Delete character at cursor

7339 *Synopsis* : [*buffer*] [*count*] `x` 8

7340 Delete *count* characters in the current line starting with the current column posi-
7341 tion; equivalent to [*buffer*] [*count*] `dl`. If the number of characters to be deleted 8
7342 is greater than or equal to the number of characters to the end of the line, delete
7343 all of the characters from the current position to the end of the line and move the
7344 cursor to the new last character on the line.

7345 *Current line* : Unchanged.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Current column : Set to the greater of (*current column* − the width of the *count* deleted characters) or 1.

5.35.7.1.83 Delete character before cursor

Synopsis : [*buffer*] [*count*] `X`

Delete the character before the cursor, equivalent to [*buffer*] [*count*] `dh`. A *count* shall repeat the effect, but only characters on the current line shall be deleted.

Current line : Unchanged.

Current column : Set to the greater of (*current column* − the width of the *count* deleted characters) or 1.

5.35.7.1.84 Yank

Synopsis : [*buffer*] [*count*] `y` *motion*

Copy (''yank'') the area of text specified by *count* and *motion* into the unnamed buffer, as well as into any named buffer specified by *buffer* .

Current line : Unchanged.

Current column : Unchanged.

5.35.7.1.85 Yank current line

Synopsis : [*buffer*] [*count*] `Y`

Place (yank) a *count* copies of the current line, equivalent to a [*buffer*] [*count*] `yy` command; see 5.35.7.1.84.

Current line : Unchanged.

Current column : Unchanged.

5.35.7.1.86 Redraw window

Synopsis : [*count1*] `z` [*count2*] *character*

Redraw the screen with a window *count2* lines long containing line *count1* placed as specified by *character* : `<newline>` specifies the top of the screen, `.` specifies the center of the screen, and − specifies the bottom of the screen. If *count1* is not specified, it shall default to the current line; if *count2* is not specified, it shall default to *window* .

Current line : *count1* , if specified; otherwise, unchanged.

Current column : Move to the first nonblank character of the current line or the first character if the line is a blank line.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.35.7.1.87 Exit

Synopsis : `ZZ`

Exits the editor, writing out the buffer if it was changed since the last write to any file. This command shall behave identically to the `ex` command `xit`; see 5.10.7.2.38.

5.35.8 Exit Status

The `vi` utility shall exit with one of the following values:

0 Successful completion.

>0 An error occurred.

5.35.9 Consequences of Errors

Default.

5.35.10 Rationale. *(This subclause is not a part of P1003.2a)*

Usage, Examples

The following table is a condensed version of information contained in the normative text. It is presented here to facilitate the review for accuracy of the `vi` commands and the editor options that affect those commands.

`vi` Command	**Editor Option**	**Description**
`control-B`	`window`	Determines screen size for paging.
`control-D`	`scroll`	Determines number of lines to be scrolled.
`control-F`	`window`	Determines screen size for paging.
`control-T`	`shiftwidth`	Move cursor forward shiftwidth positions.
`control-U`	`scroll`	Determines number of lines to be scrolled.
`{ }`	`paragraphs`	Defines additional paragraph boundaries for the `{` and `}` commands.
`[[ ]]`	`sections`	Defines additional section boundaries for the `[[` and `]]` commands.
`/`	`ignorecase`	Characters that have upper- and lowercase representations shall have those representations considered as equivalent for purposes of regular expression comparison.
	`magic`	When set, the characters `^ $ . * [ ~` have special meaning in regular expressions. When unset, only characters `^ $` have special

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

		meaning in regular expressions.
<	shiftwidth	Determines the number of character positions to be shifted.
>	shiftwidth	Determines the number of character positions to be shifted.
?	ignorecase	Characters that have upper- and lowercase representations shall have those representations considered as equivalent for purposes of regular expression comparison.
	magic	When set, the characters ^ $. * [~ have special meaning in regular expressions. When unset, only characters ^ $ have special meaning in regular expressions.
O, o	autoindent	Starting indentation determined by the line on which the command is given.
S	autoindent	Starting indentation determined by the line being changed.
ZZ	readonly	Shall not allow a write to a file.
z	window	Redraw window with the number of lines equal to the default set by window.

The control-L and control-R commands differ in their treatment of deleted ‘‘@’’lines. The former always redisplays the screen and does not affect the status of any @ line; the latter redisplays only when needed to process the deleted @ lines.

When executing an individual ex command by entering :, it is not possible to enter a <newline> as part of the command because it is considered the end of the command. As explained in 5.10.7.4, a control-M preceded by a control-V (or control-Q) escape character in a regular expression substitution is mapped into a <newline>. A different approach is to enter ex command mode by using the vi Q command (and later resuming visual mode with the ex ‘vi’’command). In ex command mode, the single-line limitation does not exist. So, for example, the following is valid:

```
Q
s/break here/break\
here/
vi
```

The uses described for control-V can also be accomplished with control-Q, which is useful on terminals that use control-V for the down-arrow function. However, most historical implementations use control-Q for the *termios* START character, so the editor will generally not receive the control-Q unless stty –ixon mode is set. Any of the command characters described in this standard can be made ineffective by their selection as *termios* control characters, using the stty utility or other methods described in POSIX .1 {8}.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

History of Decisions Made

Historically the `control-B` command skips two lines between each pair of pages it displays from a file. The working group viewed this as a bug and decided to go along with how the command was documented (as overlapping, not skipping, two lines) instead of standardizing the historical implementation.

See the Rationale for the `ex` utility for more information on `vi`.

A ‘‘restricted editor’’ (both the historical `red` utility and modifications to `vi`) were considered and rejected for inclusion. Neither option provided the level of security that many users might expect.

The `[[`, `]]`, `(`, `)`, `{`, and `}` commands are all affected by ‘‘section boundaries,’’ but in current implementations not all of the commands recognize the same section boundaries. This is considered to be a bug rather than a feature and a unique section-boundary algorithm was not described for each command.

The `e` and `E` commands described here differ slightly from some historical implementations in that an empty or blank line is described as containing exactly one word or bigword, just as for `b`, `B`, `w`, and `W`. On some systems, `E` command skips over empty/blank lines. The working group felt that a consistent definition of bigword between the `E` command and the others was sufficient justification to change one so that they all treat empty/blank lines the same way.

The `=` (reindent) command was omitted because it was LISP language-specific. (See the `ex` Rationale for the reasons LISP support was omitted.)

The text in Standard Output reflects the usage of the verb ‘‘display’’ in this clause; some `vi` implementations actually use standard output to write to the terminal, but POSIX.2 does not require that to be the case.

The action taken for signals other than SIGINT, SIGCONT, and SIGHUP is unspecified because some implementations attempt to save the editing buffer in a useful state when other signals are received.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.36 who — Display who is on the system

5.36.1 Synopsis

```
who [-mTu]
```

5.36.2 Description

The who utility shall list various pieces of information about accessible users. The domain of accessibility is implementation defined.

The default output shall contain at least the following information for accessible users, in an unspecified format: the user's login name, terminal name, and time at which the user logged in.

5.36.3 Options

The who utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

-m Output only information about the current terminal.

-T Show the state of each terminal, as described in 5.36.6.1.

-u Output each displayed user's ''idle time'' in addition to any other information. The idle time is the time since any activity occurred on the user's terminal. The method of determining this is unspecified. 8

5.36.4 Operands

None.

5.36.5 External Influences

5.36.5.1 Standard Input

None.

5.36.5.2 Input Files

None.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.